# Explanation of the Python Code for Predictive Analysis

This document provides a detailed explanation of the Python program designed for predictive analysis. The program loads data from an SQLite database, preprocesses the data, evaluates regression models, predicts outcomes, and calculates relationships between independent and dependent variables. All outputs are redirected to a DataFrame for better traceability.

## 1. Importing Required Libraries

The following libraries are imported for various tasks:

* **sqlite3**: Connects to the SQLite database.
* **pandas**: Handles data manipulation and storage.
* **numpy**: Performs numerical operations.
* **scikit-learn**: Implements machine learning models and preprocessing techniques.
* **matplotlib**: Plots graphs.

## 2. Function Descriptions

### 2.1 load\_data(db\_path, table\_name)

**Purpose**: Loads data from an SQLite database table.

**Parameters**:

* db\_path: Path to the SQLite database file.
* table\_name: Name of the table to fetch data from.

**Returns**: A DataFrame containing the data or an error message in case of failure.

### 2.2 preprocess\_data(data)

**Purpose**: Preprocesses the loaded data by:

* Converting dates into separate columns (Year, Month, Day).
* Encoding categorical variables.
* Imputing missing values in numerical columns.

**Parameters**:

* data: The raw data as a DataFrame.

**Returns**: The processed DataFrame and a log of preprocessing steps.

### 2.3 choose\_best\_regression\_method(X, y)

**Purpose**: Evaluates multiple regression models using cross-validation and selects the best-performing one.

**Models Evaluated**:

* Random Forest Regressor
* Linear Regression
* Ridge Regression
* Lasso Regression

**Parameters**:

* X: Features (independent variables).
* y: Target (dependent variable).

**Returns**:

* The best model.
* A DataFrame with cross-validation scores for each model.

### 2.4 train\_model(X, y, model)

**Purpose**: Trains the selected regression model on the entire dataset.

**Parameters**:

* X: Features.
* y: Target.
* model: The regression model to be trained.

**Returns**: The trained model.

### 2.5 get\_significant\_parameters(model, feature\_columns)

**Purpose**: Identifies significant parameters affecting predictions based on feature importance.

**Parameters**:

* model: The trained regression model.
* feature\_columns: List of feature column names.

**Returns**: A DataFrame with feature importance values or a message if the model does not support feature importance.

### 2.6 predict\_next\_six\_months(model, data, feature\_columns, target)

**Purpose**: Predicts the target variable for the next six months (January 2025 to June 2025) for all factories and locations.

**Parameters**:

* model: The trained regression model.
* data: The original dataset.
* feature\_columns: List of feature column names.
* target: The target variable to predict.

**Returns**: A DataFrame with predictions for the next six months.

### 2.7 calculate\_relationships(data, independent\_columns, dependent\_column)

**Purpose**: Calculates correlations between independent and dependent variables.

**Parameters**:

* data: The dataset.
* independent\_columns: List of independent variables.
* dependent\_column: The dependent variable.

**Returns**: A DataFrame with correlation values for each feature.

### 2.8 interactive\_prediction()

**Purpose**: Handles the entire workflow interactively, including:

1. Loading data.
2. Preprocessing.
3. Evaluating regression models.
4. Training the best model.
5. Identifying significant parameters.
6. Predicting future outcomes.
7. Calculating correlations.

**Details**:

* Works with a predefined SQLite database path (db\_path) and table name (table\_name).
* Iterates through multiple target variables (e.g., Production Volume, Revenue, Foam Density).
* Redirects all outputs to a final DataFrame and saves them as a CSV file (output\_results.csv).

## 3. Workflow Execution

The interactive\_prediction() function is executed in the \_\_main\_\_ block, which:

1. Loads data from the SQLite database.
2. Preprocesses the data.
3. Selects the best regression model.
4. Trains the model.
5. Predicts the target variable for the next six months.
6. Calculates relationships between variables.
7. Redirects all outputs to a DataFrame and saves the results to output\_results.csv.

## 4. Output Details

All outputs (logs, scores, predictions, and correlations) are collected in a single DataFrame for better traceability. This DataFrame is exported to a CSV file named output\_results.csv.

## 5. Key Advantages of the Code

1. **Modular Design**:
   * Each function performs a specific task, ensuring reusability and clarity.
2. **Scalability**:
   * Can handle multiple target variables and large datasets.
3. **Traceability**:
   * Outputs are logged into a DataFrame and saved for future analysis.
4. **Interactive Workflow**:
   * Automates the process of model evaluation, training, and prediction.
5. **Predictive Analysis**:
   * Provides insights into future outcomes and significant parameters.

## 6. Summary

This Python program is a comprehensive solution for predictive analysis. It automates data loading, preprocessing, model evaluation, and future prediction, ensuring outputs are traceable and exportable for further use. The modular design and output redirection to a DataFrame make the program robust and user-friendly.